Amendments to the Claims

Please amend claims 6, 12 and 16. A complete listing of claims all and status indicators is set forth below. This listing of claims will replace all prior versions and listings of claims in the application.

1. (original) A method for storing transaction entries in a transaction order queue, comprising the acts of:

a) temporarily storing a plurality of transaction entries;

b) selecting one of the plurality of temporarily stored transaction entries; and

c) enqueuing the selected one of the plurality of temporarily stored transaction entries in the transaction order queue.

2. (original) The method as recited in claim 1, wherein act (a) comprises the act of temporarily storing a plurality of transaction entries in a bank of registers.

3. (original) The method as recited in claim 1, wherein the plurality of transaction entries are temporarily stored simultaneously.

4. (original) The method as recited in claim 1, wherein acts (b) and (c) comprise the acts of:

determining whether a posted write transaction entry is present;

if the posted write transaction entry is present, then enqueuing the posted write transaction entry into the transaction order queue, if the posted write transaction entry is not present, then determining whether a read completion transaction entry is present;

if the read completion transaction entry is present, then enqueuing the read completion transaction entry into the transaction order queue;

if the read completion transaction entry is not present, then determining whether a delayed/split transaction entry is present; and

if the delayed/split transaction entry is present, then enqueuing delayed/split transaction entry into the transaction order queue.

5. (original) The method as in claim 1, comprising the act of enqueing each of the plurality of transaction entries into the transaction order queue one at a time during successive clock cycles.

6. (currently amended) A method of manufacturing a computer system for storing transaction entries in a transaction order queue comprising the acts of:

providing a temporary storage space for temporarily storing a plurality of transaction entries;

providing logic to select one of the plurality of temporarily stored transaction entries; and providing a transaction order queue to receive the selected one of the plurality of temporarily stored transaction entries.

7. (original) The method as recited in claim 6, wherein the act of providing temporary storage space for temporarily storing a plurality of transaction entries comprises the act of providing a plurality of registers.

8. (original) The method as recited in claim 7, wherein the act of providing logic to select one of the plurality of temporarily stored transaction entries comprises the act of providing logic for:

determining whether a posted write transaction entry is present;

if the posted write transaction entry is present, then enqueuing the posted write transaction entry into the transaction order queue;

if the posted write transaction entry is not present, then determining whether the transaction is a read completion transaction entry;

if the read completion transaction entry is present, then enqueuing the read completion transaction entry into the transaction order queue;

if the read completion transaction entry is not present, then determining whether a delayed/split transaction entry is present; and

if the delayed/split transaction entry is present, then enqueuing the delayed/split transaction entry into the transaction order queue.

9. (original) An system for providing multiple simultaneous transaction entries to a transaction order queue comprising:

means for temporarily storing simultaneously a plurality of transaction entries simultaneously;

means for selecting one of the plurality of temporarily stored transaction entries; and

means for enqueuing the selected one of the plurality of temporarily stored transaction entries in a transaction order queue.

10. (original) The system as in claim 9, wherein the means for temporarily storing a plurality of transaction entries comprises a bank of registers.

11. (original) The system as in claim 9, wherein the means for selecting one of the plurality of temporarily stored transaction entries comprises:

means for determining whether a posted write transaction entry is present;

means for enqueuing the posted write transaction entry into the transaction order queue if the posted write transaction entry is present;

means for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

means for enqueuing the read completion transaction entry into the transaction order queue if the read completion transaction entry is present;

means for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

means for enqueuing the delayed/split transaction entry into the transaction order queue if the delayed/split transaction entry is present.

12. (currently amended) A system for providing multiple simultaneous transaction entries to a transaction order queue comprising:

a temporary memory storage adapted to store a plurality of transaction entries; and

logic adapted for selecting and ordering the plurality of transaction entries in the transaction order queue from the temporary memory storage for processing.

13. (original) The system as in claim 12, wherein the temporary memory storage comprises a bank of registers.

14. (original) The system as in claim 12, wherein the logic comprises:

logic adapted for determining whether a posted write transaction entry is present;

logic adapted for enqueuing the posted write transaction entry into the transaction order queue if the posted write transaction entry is present;

logic adapted for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

logic adapted for enqueuing the read completion transaction entry into the transaction order queue if the read completion transaction entry is present;

logic adapted for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

logic adapted for enqueuing the delayed/split transaction entry into transaction order queue if the delayed/split transaction entry is present.

15. (original) The method as in claim 12, wherein each of the plurality of transaction entries is enqueued into the transaction order queue one at a time during successive clock cycles.

16. (currently amended) A ~~An~~ processing system comprising:

a first logic device;

a plurality of registers being configured to receive a plurality of transaction entries from the first logic device ~~as ordered by the first logic device~~;

a second logic device adapted to receive the transaction entries from the plurality of registers and being programmed to select transaction entries according to PCI-X specifications; and

a transaction order queue being configured to receive and enqueue the selected transaction entries.

17. (original) The processing system as in claim 16, wherein the first logic device receives transaction entries from an input source.

18. (original) The processing system as in claim 16, wherein the plurality of registers store the plurality of transaction entries received from the first logic device.

19. (original) The processing system as in claim 16, wherein the second logic device selects a single entry to send to the transaction order queue.

20. (original) The processing system as in claim 19, wherein the second logic device comprises:

logic adapted for determining whether a posted write transaction entry is present;

logic adapted for enqueuing the posted write transaction entry into the transaction order queue if the posted write transaction entry is present;

logic adapted for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

logic adapted for enqueuing the read completion transaction entry into the transaction order queue if the read completion transaction entry is present;

logic adapted for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

logic adapted for enqueuing the delayed/split transaction entry into the transaction order queue if the delayed/split transaction entry is present.

21. (original) The processing system as in claim 16, wherein the transaction order queue enqueues one transaction entry per clock cycle.

22. (original) A computer system comprising:

at least one processor;

a memory device operatively coupled to the at least one processor; and

a transaction order queue circuit configured to process transactions from the memory device, the transaction order queue circuit being adapted to encode a plurality of simultaneous transaction entries.

23. (original) The computer system as in claim 22, wherein the transaction order queue comprises:

logic adapted for determining whether a posted write transaction entry is present;

logic adapted for enqueuing the posted write transaction entry into the transaction order queue, if the posted write transaction entry is present;

logic adapted for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

logic adapted for enqueuing the read completion transaction entry into the transaction order queue, if the read completion transaction entry is present;

logic adapted for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

logic adapted for enqueuing the delayed/split transaction entry into the transaction order queue if the delayed/split transaction entry is present.

24. (original) The system as in claim 22, wherein the computer system comprises network capabilities.

25. (original) A method for storing transaction entries in a transaction order queue, comprising the acts of:

a) temporarily storing a plurality of simultaneous transaction entries; and

b) delivering the plurality of transaction entries to a transaction order queue one at a time.

26. (original) The method as in claim 25, wherein the plurality of simultaneous transaction entries is stored in a bank of registers.

27. (original) The method as in claim 25, wherein act (b) comprises the act of:

determining whether a posted write transaction entry is present;

enqueuing the posted write transaction entry into the transaction order queue, if the posted write transaction entry is present;

determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

enqueuing the read completion transaction entry into the transaction order queue, if the read completion transaction entry is present;

determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

enqueuing the delayed/split transaction entry into the transaction order queue.

28. (original) A method for storing transaction entries in a transaction order queue, comprising the acts of:

a) temporarily storing a plurality of simultaneous transaction entries;

b) prioritizing each of the temporarily stored transaction entries; and

c) transmitting the stored transaction entries to the transaction order queue according to priority.

29. (original) The method as in claim 28, wherein the plurality of transaction entries are stored simultaneously in a bank of registers.

30. (original) The method as in claim 28, wherein act (b) comprises the acts of:

determining whether a posted write transaction entry is present;

enqueuing the posted write transaction entry into the transaction order queue, if the posted write transaction entry is present;

determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

enqueuing the read completion transaction entry into the transaction order queue, if the read completion transaction entry is present;

determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

enqueuing the delayed/split transaction entry into the transaction order queue.